

Scribe3 deployment RFC

Last updated at 7/30/15 5:56 PM

Introduction and motivation

This document describes a proposed process for the deployment and update of TT-Scribe3 software. The purpose is to formalize the standards and pathways by which software is delivered to the scribes. This is a RFC, and thus none of the concept hereby expressed should be considered final until approved as such.

Objectives

The final objective is to have a single, standardized procedure where the receiver of a TT-Scribe has to do a minimal amount of non-technical work to get the system up and running, and to maintain it to the latest update. In this context I propose “non-technical” be loosely interpreted as “inside the boundaries of the Scribe3 application”, as in: all the necessary configuration and update operations that do require user input must be exposed by the Scribe3 software and no command-line action should be required by the user at all.

Scope

Deployment procedures apply to TT-Scribes operated both by Archive, as well as third party customers, both in production and not, running on either Ubuntu 12.04.5 LTS or 14.04 LTS.

Scenarios

Deployment procedures and concerns differ whether the TT-Scribe is currently used in production or not, and whether it will be. The following table describes the deployment and update strategies, as well as concerns and specifics for each scenario.

<i>Scenario</i>	<i>Concerns</i>	<i>Specifics</i>	<i>Deployment</i>	<i>Update</i>
<u>Production scribes</u>	<ul style="list-style-type: none">• Conserve the local book library• Maximize up & running time• Guarantee the ability of local support to fix issues	<ul style="list-style-type: none">• No direct physical access• No remote access to machine• Various versions	In case re-deployment is required, the suggested way is through books-yaz, re-generating the install bash script.	Updates pulled manually from Debian repository by system maintainer with apt-get
<u>New scribes</u>	<ul style="list-style-type: none">• Delivering a working product out-of-the-box• Minimize the barriers and time to up & running• Ensure availability of strategies to deal with software issues	<ul style="list-style-type: none">• Initial installation is handled by Archive team• Archive account needs to be created for s3 access	<ul style="list-style-type: none">• Need to download a script from Cluster Republisher and run it locally, with possibly some extra system administration needed.• Metadata files need to be setup manually	Updates pulled manually from Debian repository by system maintainer
<u>Test scribes</u>	<ul style="list-style-type: none">• Provide actionable feedback to devs• Ensure testing environment is realistic and representative of production scribes• Provide rollback strategy	Same as production scribes	Not applicable	Updates are pulled manually from ephemeral repositories on archive.org and manually performs the upgrade.

Course of action

Issues and requirements

In this section the main issues and relative requirements are listed.

- Complex, custer-rep dependent deployment based on a bash script. This procedure should be deprecated all together in favor of an automated configuration procedure that can be carried out entirely on the TT-Scribe, and the staples of which would be:
 - OS installation (Ubuntu 14.04)
 - ia-scribe bundle installation
 - User login and initial metadata configuration (see S3 keys paragraph below)
- User needs to interact with system files to edit metadata. Both in terms of initial configuration, as well as in current use instances, the user should be able to do change books and scribe's metadata from inside the application.
- **S3 keys are bundled.** Authentication to Archive's backend is granted though the use of S3 keys. Although the user is not currently required to insert them, as they get shipped as part of the aforementioned bash script, a panel allows for their modification. In order not to expose this information, as well as to break free from the bash script deployment, authentication should be dealt with by logging in to Archive with username/password.
- **Disaster recovery/rollback procedure are not present.** In case an update or upgrade break the system, it should be possible to revert to a working state quickly and without data loss.
- **OS is outdated.** TT-scribes should ship with Ubuntu 14.04 LTS by default.

Tasks and prioritization

In order to deliver on the requirements presented above, the following task prioritization is proposed:

1. Trusty upgrade: The first step in streamlining the delivery of TT-Scribes is to provide a solid installation base on Ubuntu 14.04 LTS. This means that the software must be made compatible with Trusty both in the case of an **update** from 12.04.05 or from a **fresh** installation. Currently, update is supported and tested working. Some issues remain in providing the correct libraries for a fresh installation.
2. Metadata editing: This issue has priority over the archive login because it'd benefit scribes already in production.
3. **Archive login:** There has been some discussion about how to do this. It should be possible to use the `internetarchive` python package, but a detailed design hasn't been decided yet. There are also some open questions (see below).
4. **Disaster recovery:** This requirement needs to be validated and further broken down before a valid priority can be established.

Time estimate and risk profiles

Delivery of the first three features listed above should take place within August 2015. In particular, a Trusty upgrade procedure for clean system install should be available within the first week of August. I estimate the cost of the metadata editing and archive login in about ten days each.

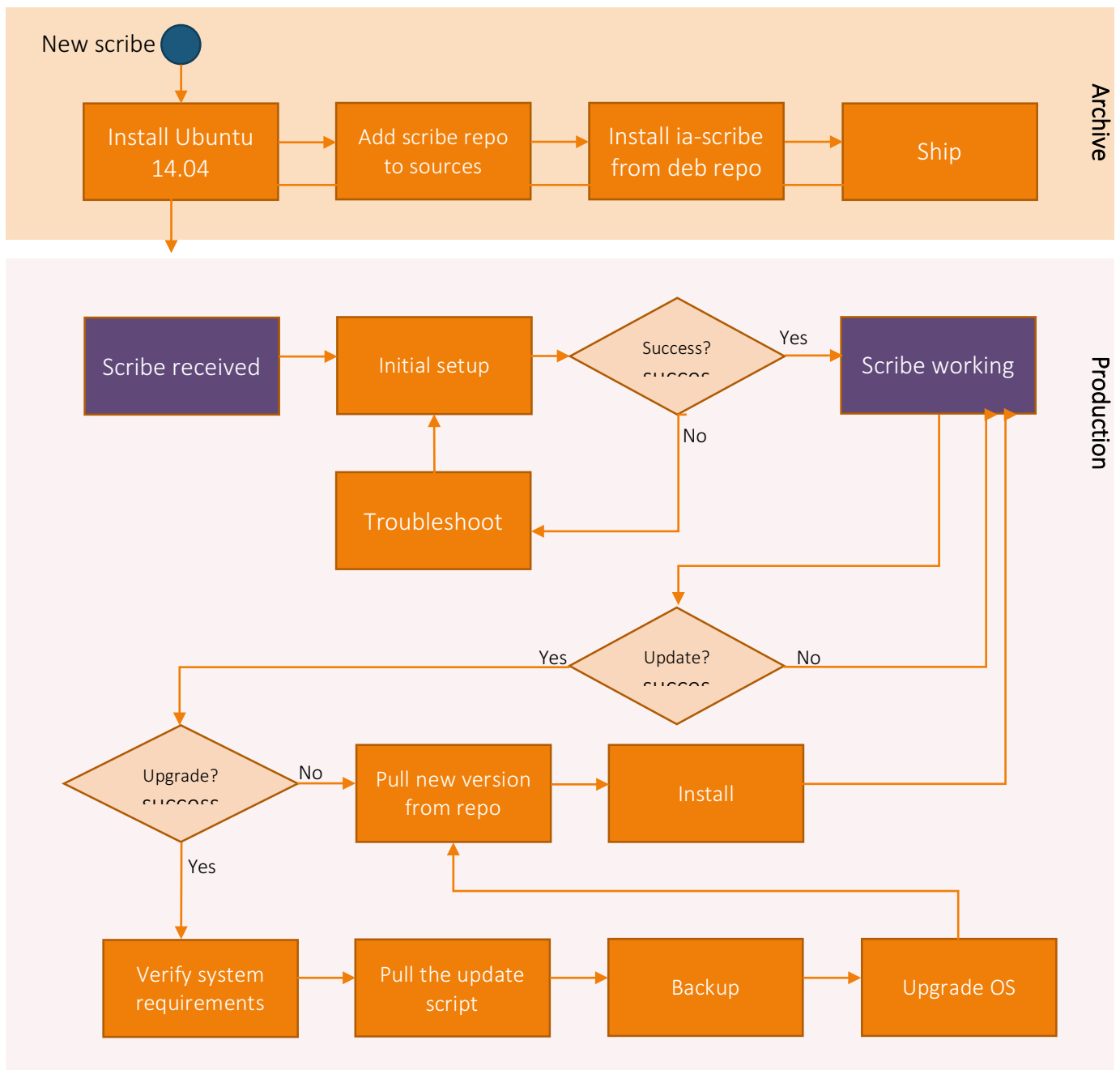
The main risk is, as usual, runaway development cost due to the writhed nature of the software and of the components around it (libraries and OS). A good mitigation strategy is to proceed by micro-requirements trying to maintain working versions, limit refactoring as much as possible and limit intervention to isolated

components. Additional risks derive from the lack of tests, documentation and support; they have been mitigated so far only by means of costly code analysis.

Open questions

- What workflow to use to apply Cluster Rep access privileges to a certain user account? (do we create it? Do they create it and then notify us? Can it be automated?)
- Should Scribe3 retain the access keys or delegate login entirely to the internetarchive pip package? Is there a need for multi-user support?
- What should disaster/rollback recovery entail: what to back up and where to back it up?

Overall desired process overview



Legend

- **Purple:** States
- **Orange:** actions